Name: Sahib Bajwa

ID: 107553096

CSCI 3104, Algorithms **Profs. Chen & Grochow**

Final Exam S9–S11 **Spring 2020, CU-Boulder**

Instructions: This quiz is open book and open note. You **may** post clarification questions to Piazza, with the understanding that you may not receive an answer in time and posting does count towards your time limit. Questions posted to Piazza **must be posted as PRIVATE QUESTIONS.** Other use of the internet, including searching for answers or posting to sites like Chegg, is strictly prohibited. Violations of these are grounds to receive a 0 on this quiz. Proofs should be written in **complete sentences. Show and justify all work to receive full credit.**

TIMING: If you are not attempting all the standards in a given quiz, please only use the ordinary amount of time for the number of standards you attempt. For example, if you are only attempting one standard on a 4-standard quiz, please only use 30 min (or 38 for 1.5x, 45 for 2x).

YOU MUST SIGN THE HONOR PLEDGE. Your quiz will otherwise not be graded.
Honor Pledge: On my honor, I have not used any outside resources (other than my notes and book), nor have I given any help to anyone completing this assignment.

Your Name: Sahib Bajwa

Quicklinks: 9 9a 9b 10 10a 10b 10c 10d 11 11a 11b

---

9. **Standard 9.** *Suppose that we modify the* Partition *algorithm in QuickSort in such a way that on even levels of the recursion tree,* Partition *chooses the best possible pivot, and on odd levels of the recursion tree,* Partition *chooses the last element as the pivot. You may assume that, as with ordinary QuickSort,* Partition *still takes only $O(n)$ steps to run on a list of length n.*

   THIS QUESTION HAS TWO PARTS, (a) and (b).

   (a) *What would be the worst-case running time of the resulting QuickSort algorithm? Use a suitable asymptotic notation to represent the running time, and* **explain/justify** *your answer.*

   (BEGIN YOUR ANSWER ON THE NEXT PAGE.)

The worst-case running time of the resulting QuickSort algorithm will be $\Theta(n^2)$. This is because of the way that this version of QuickSort operates. For half of the algorithm, we will choose the best pivot. The other half of the time during the worst-case, we will chose the worst pivot. The best-pivot time will be $\Theta(nlogn)$. The worst-pivot time will be $\Theta(n^2)$. Together, this will be $\Theta(n^2) + \Theta(nlogn)$. Simplifying, we know through the limit test that $nlogn \leq O(n^2)$. Thus, the worst-case running time of the resulting QuickSort algorithm will be $\Theta(n^2)$.

I also believe that since we have a height of at least $O(n/2)$ where the running time is of the worst case, we will lead to a time of $O(n^2/2)$.

(b) *What would be the average running time of the resulting QuickSort algorithm? Use a suitable asymptotic notation to represent the running time, and* **explain/justify** *your answer.*
(BEGIN YOUR ANSWER ON THE NEXT PAGE.)

The average running time of the resulting QuickSort algorithm will be $\Theta(nlogn)$. For half of the the algorithm, we will chose the best pivot. For the other half, we do not know what we will get, or we will get the average. The best-pivot time will be $\Theta(nlogn)$. The average-pivot time will be $\Theta(nlogn)$. Since they are both the same, the entire algorithm will have a running time of $\Theta(nlogn)$. Thus, the average running time of the resulting QuickSort algorithm will be $\Theta(nlogn)$.

10. **Standard 10.** *Consider the hash function $h(k) = \lfloor 10k \rfloor$ for all keys k for a table of size 10. (Resolve collisions by chaining with a linked list.) You have three applications:*

    - *Application 1: keys are randomly drawn from the interval* $[0, 0.5]$
    - *Application 2: keys are randomly drawn from the intervals* $[0, 0.2] \cup [0.4, 0.6] \cup [0.8, 1]$
    - *Application 3: keys are randomly drawn from the interval* $[0, 1]$.

    THIS QUESTION HAS FOUR PARTS, (a)–(d).

    (a) *For which application does the hash function $h(k)$ perform better? Please* **explain/justify adequately** *your answer.*
    (BEGIN YOUR ANSWER ON THE NEXT PAGE.)

YOUR ANSWER HERE FOR 10a. (YOU CAN DELETE ALL THIS TEXT IN CAPS.)

IF YOU ARE HANDWRITING AND INSERTING AN IMAGE, SEE THE COMMENTED CODE BELOW IN THE .TEX FILE. PLEASE BE SURE TO ROTATE YOUR IMAGE TO THE CORRECT ORIENTATION (CAN BE DONE IN THE LATEX DIRECTLY; SEE COMMENTS.)

(b) *In each of the three applications, does the hash function $h(k)$ satisfy the uniform hashing property? Please* **explain/justify adequately** *your answer.*
(BEGIN YOUR ANSWER ON THE NEXT PAGE.)

YOUR ANSWER HERE FOR 10b. (YOU CAN DELETE ALL THIS TEXT IN CAPS.)

IF YOU ARE HANDWRITING AND INSERTING AN IMAGE, SEE THE COMMENTED CODE BELOW IN THE .TEX FILE. PLEASE BE SURE TO ROTATE YOUR IMAGE TO THE CORRECT ORIENTATION (CAN BE DONE IN THE LATEX DIRECTLY; SEE COMMENTS.)

(c) *Suppose you have n keys in total for each application. What is the resulting load factor for each application?*

(BEGIN YOUR ANSWER ON THE NEXT PAGE.)

YOUR ANSWER HERE FOR 10c. (YOU CAN DELETE ALL THIS TEXT IN CAPS.)

IF YOU ARE HANDWRITING AND INSERTING AN IMAGE, SEE THE COMMENTED CODE BELOW IN THE .TEX FILE. PLEASE BE SURE TO ROTATE YOUR IMAGE TO THE CORRECT ORIENTATION (CAN BE DONE IN THE LATEX DIRECTLY; SEE COMMENTS.)

(d) *Suppose you have n keys in total for each application. What are the time complexities of the dictionary operations: add, delete, and search, respectively?*
(BEGIN YOUR ANSWER ON THE NEXT PAGE.)

YOUR ANSWER HERE FOR 10d. (YOU CAN DELETE ALL THIS TEXT IN CAPS.)

IF YOU ARE HANDWRITING AND INSERTING AN IMAGE, SEE THE COMMENTED CODE BELOW IN THE .TEX FILE. PLEASE BE SURE TO ROTATE YOUR IMAGE TO THE CORRECT ORIENTATION (CAN BE DONE IN THE LATEX DIRECTLY; SEE COMMENTS.)

11. **Standard 11.** Consider the interval scheduling problem from class. You are given a list of intervals, each with a start and end time (s_i, e_i). Your goal is to select a subset S of the given intervals such that (a) no two intervals in S overlap, and (b) S contains as many intervals as possible subject to condition (a).

    A comment on level of justification for Q11: Write a little about the order in which the algorithm selects the intervals. For example "The algorithm takes intervals in the order [1,3,4]: first the algorithm takes interval 1 because that is the shortest. Interval 1 conflicts with intervals 2 and 5, so they are removed. The next shortest is 3, which conflicts with interval 6, and the only remaining interval is 4."

    THIS QUESTION HAS TWO PARTS, (a) and (b).

    (a) For each interval (s_i, e_i), define a function

    $$f(s_i, e_i) = (e_i - s_i) \times e_i$$

    Note that this function is generally smaller for intervals that are shorter and/or end earlier. Consider a greedy algorithm which always selects the appointment which minimizes f first (then removes conflicting intervals, then repeats). Draw an example where this algorithm fails. List the order in which the algorithm selects the intervals, and also write down a larger subset of non-overlapping intervals than the subset output by the greedy algorithm.
    (BEGIN YOUR ANSWER ON THE NEXT PAGE.)

YOUR ANSWER HERE FOR 11a. (YOU CAN DELETE ALL THIS TEXT IN CAPS.)

IF YOU ARE HANDWRITING AND INSERTING AN IMAGE, SEE THE COMMENTED CODE BELOW IN THE .TEX FILE. PLEASE BE SURE TO ROTATE YOUR IMAGE TO THE CORRECT ORIENTATION (CAN BE DONE IN THE LATEX DIRECTLY; SEE COMMENTS.)

(b) Consider the greedy algorithm which selects the interval with the earliest start time s_i first (then removes conflicting intervals, then repeats). Draw an example to show that this algorithm is not optimal. Show what this algorithm does on your example, what subset it outputs, and show a strictly larger set of non-overlapping intervals.

(BEGIN YOUR ANSWER ON THE NEXT PAGE.)

YOUR ANSWER HERE FOR 11b. (YOU CAN DELETE ALL THIS TEXT IN CAPS.)

IF YOU ARE HANDWRITING AND INSERTING AN IMAGE, SEE THE COMMENTED CODE BELOW IN THE .TEX FILE. PLEASE BE SURE TO ROTATE YOUR IMAGE TO THE CORRECT ORIENTATION (CAN BE DONE IN THE LATEX DIRECTLY; SEE COMMENTS.)